

Databázové systémy

palovska@vse.cz

# *Další zdroje k předmětu*

- <http://nb.vse.cz/~palovska/uds>
  - <http://nb.vse.cz/~palovska/uds/uds.ppt>
  - <http://nb.vse.cz/~palovska/uds/tskripta.zip>
    - spíše k bakalářskému základu o DB

# Co je databáze?

- Mnoho dat
- Organizovaných
  - jak?
- Ochrana a správa přístupu
- Kdo co = role
- Aplikace, jejich typy
- Jak se to dělá (od začátku)

# *Jak jsou data organizovaná?*

- Co je relační model...
- Co bylo předtím? (A je to pořád.)
- Co následovalo po relačním
  - Objektové db
  - „Objektově relační“
- Jiný směr
  - Multidimenzionální (OLAP kostky)
  - XML
  - Proudý dat

# *Relační model*

- Relace = tabulky
- Pole = sloupce
  - Datové typy
  - Sémantika: název, popis
  - Doména?
- Řádky = záznamy
  - Identita
  - Vztahy mezi řádky...
- Cizí klíče = základ relačního modelu

# Co znamená relační model

- Mnohost, vícenásobnost vlastní entitám
  - je začarována do obtížných sestav tabulek
- Hierarchické složení
  - je buď rozloženo a zanedbáno
  - nebo je začarováno do mnoha cizích klíčů
- Identita závisí na sémantickém „primárním“ klíči
- Realizace relačního modelu (implementace v DBMS)
  - způsobila chudost výběru efektivních „datových typů“

# *Objektové principy v DB*

- Objektová identita
  - ID záznamu objektu, jedinečný a nevýznamový
- Vztahy
  - pomocí REF jakožto atributu
  - vícenásobný vztah jako SET(REF...)
  - (nejsou potřeba cizí klíče?)
- Komplexní objekt zaznamenán jako celek, rozklad je sice možný, ale ne nutný
- V principu otevřená množina „typů“

# *Objektová identita*

- „Primární klíče“ jen pro vyhledávání, není třeba „primární“, stačí „klíče“
- Vztahy pomocí OID a REF
- Důsledky:
  - snazší formulace požadavku na související data
  - kolize z nepořádku v aplikační oblasti není v db fatální
    - za „klíče“ zodpovídají lidé



# *Mnohost, vícenásobnost*

- Konstrukce SET, MULTISSET, LIST, ARRAY
- Jazyk pro náležení do množiny
  - IN
- U vztahů reverzní vztah, konzistenci má (?)  
hlídat DBS
  - může být též SET... (n:m)

# *Celek a složky*

- Konstrukce STRUCT
  - každá složka má jméno
  - ve všech úrovních
- Možno rozložit a složit něco jiného v dotazech

# *Manipulace s objekty*

- Objektově orientovaným jazykem
  - uložit
  - změnit hodnotu
  - zavolat metodu třídy
- DSMS mu rozumí
  - netřeba middleware, frameworky...

# Zapouzdření

- Vlastník objektu mění hodnotu
- Ostatní volají metody, pokud k nim mají oprávnění
- Dotazy:
  - možno porušovat zapouzdření!
  - informace o datové struktuře je veřejná
  - čtení je řízeno a využívá systému práv

# *Dědičnost*

- Úložiště pro nadtržidy i podtržidy
  - dotazy pro celou hierarchii
- Systém tříd persistentních i transientních
  - stejný jazyk manipulace, dotazů

# „Datové“ typy

- Základní i další nabízené DBMS
- UDT
  - jejich efektivita je závislá na kvalitě developera  
a
  - možnostech DBMS

# Čisté OODBMS

- standard ODMG 3.0
  - nikde zcela nerealizován
  - pouze ideový rámec, nikoli syntaxe
- objektový model OMG
  - třídy, stavy, metody
  - vztahy
- bindingy k OO jazykům
  - OO programování perzistence je OO nativní

# ***OODBMS***

- specifické aplikační oblasti



## 2. soustředění

# ***Datamining***

# *Podmínky nasazení*

- Efektivně uspořádaná konzistentní data
  - odstraněné nedostatky, konsolidované
- Znalost problematiky
- Představa o tom, co se chce zjistit, najít
- Kvalifikace v dataminingu
- Dataminingové nástroje

# *Proces dataminingu*

- Trénovací data
- Ověřovací data

# *Metody*

- Rozhodovací stromy
- Asociační pravidla
- Neuronové sítě
- Regresní analýza
- Shluková analýza

# *„Ruční“ datamining*

- Vhodné SQL dotazy
- +
- grafická vizualizace

# ***Hierarchické uspořádání***

# ***Příklad 1 – subordinance***

Každý žoldák má bezprostředně nadřízeného, potřebujeme sledovat linie subordinance.



## ***Příklad 2 - kusovník***

Kosmická loď se skládá z mnoha dílů, jež mohou samy být složeny z dalších dílů. Je nutno, aby bylo možno sledovat, z čeho se který díl skládá, v libovolné úrovni kompozice.

# Co zachycuje

- Celek – část:  
nadřazený element obsahuje elementy jemu podřízené
  - širší – užší pojem (množina – podmnožina)
  - množina – prvek (prvky různého typu)
- Nadřízený – podřízený:  
nadřazený element zodpovídá za elementy jemu podřízené
- Předchůdce – následník:  
kauzální následnost

# *Přístupové cesty*

- Od kořene dolů
  - Využití při klasifikaci
  - Nadřazený prvek zná cesty k jemu podřízeným
- Rekurze
  - Začít se dá kdekoli
- Přímý přístup podle ID, hodnoty
  - Indexy... (ev. hierarchické)

# *Příklady*

- Fyzická pošta, územní správa
- Internetové domény
- Firemní organizace
- File system
- LDAP
- Textové dokumenty
- XML

# *Textové databáze*

- Dokument
  - titul
  - autor, ...
  - klíčová slova
  - kapitola
    - subkapitola
      - odstavec
        - věta
          - slovo

# *Vyhledávání v textových db*

- Podle struktury (dokumenty spořádaně strukturované)
- Podle termů
- WWW
  - Jak to dělá Google?
  - Sémantický web...?

# *XML dokumenty*

- Relativně čitelné pro člověka (ukecaný)  
(samovysvětlujícínost má ale své meze)
- Doporučené W3C
- Otevřený volný standard
- Parsování standardními parsery
  - ověřování
  - zpracování

# *Kritika XML databází*

- Hierarchická struktura
  - vs. vztahy n:m
    - obtížná normalizace
  - vs. jiné „pohledy“
  - relativismus designu
- Zmatečné konstrukty
  - namespaces
  - atributy



# Výhody XML databáze

- Organizace dat odpovídající přenosovému formátu
- Snadná adaptace schématu na změny
  - oproti relačnímu modelu
- Komplexní validace
  - Kompletnost a správnost dokumentu
  - Pravidla pro „datové typy“ (regulární výrazy...)
- Otevřené standardy XML (~SQL)
  - XPath, XQuery

# ***Databáze a časový faktor***

# ***Příklad***

Navrhněte databázi pro aplikaci vyhledávání spojení (vlaky, autobusy, letadla, MHD).

# *Pojmy související s časem*

- Předchozí, následující
- Časový interval
- Historie
- Současnost

# ***Klasický databázový přístup***

- Databáze si uchovává celou historii

# *Měřicí přístroj*

V tabulce T máme pole cas a pole cis\_udaj, cas je UNIQUE.

Najděte všechny případy, kdy bezprostředně časově následující cis\_daj je nižší než jeho předchůdce.

(Kdy došlo k poklesu teploty?)

# *Měřicí přístroj – SQL*

```
SELECT t1.cas, t1.cis_udaj,t2.cas, t2.cis_udaj  
FROM T t1, T t2  
WHERE t1.cas<t2.cas  
AND NOT EXISTS (SELECT * FROM T t3  
WHERE t1.cas<t3.cas AND t3.cas<t2.cas)  
AND t1.cis_udaj>t2.cis_udaj;
```

# *Kde je chyba?*

Nutnost použít konstrukce EXISTS je způsobena nevhodným návrhem tabulky T.

Pokud by obsahovala sloupec s číslem měření, úloha by byla rázem velmi jednoduchá.



# *Více sledovaných veličin?*

- Přichází údaj o času, komoditě a její ceně.  
Řešíme úlohu „průběh cen“  
- kdy došlo k jaké změně ceny které komodity?
- Číslování přicházejících „záznamů“ nepomůže...
- Je třeba nalézt následující „záznam“ pro danou komoditu.

# Více sledovaných veličin

- Seřadíme:

↓                      ↓  
komodita    čas    cena

- Řešení podobné jako v předchozím případě, zpracujeme sériově
  - možno i v tabulkovém kalkulátoru...
- Problém tohoto řešení: potřebujeme paměť pro celou tabulku.
- =dávkové zpracování celé historie

# Řešení 2

- Pamatujme si poslední cenu pro každou komoditu, jak data přicházejí
- Nepotřebujeme znát vše najednou, jen to poslední
- Sledujeme „proud dat“

# *Proudy dat*

- Nutná paměť jen pro potřebou část fronty přicházejících dat
- Zpracování „dotazů“
- Oznamování událostí
  
- Je to databázové zpracování?
  - Zprávy nebo události můžeme ukládat do skladu
    - pro učení

# ***XML data***

- Dokument jako proud

# Časová razítka, verze

- Časové razítko vzniku/změny záznamu
  - Čas vzniku – předchozí problémy (průběh veličin?)
  - Čas změny – co bylo před změnou?
- Verze (co bylo před změnou)
  - Možný návrat k předchozím stavům
  - Formálně stejné jako při „času vzniku“, ale jiná interpretace, jiný interface

# *Transientní data*

- Omezení životnosti záznamů
  - Jak hlídat, ošetřovat rušení prošlých?
    - diskrétní vs. spojitý čas
- Události způsobené časem

# *Události způsobené časem*

- Posílání upomínek po vypršení lhůty
  - Lze triggerem?



# *Události způsobené časem*

- Zpracování v diskrétních časech

nebo

- Stálý démon, pracující v malých cyklech
  - pouze dávkové zpracování

# *Zpracování časového faktoru*

- Sériové
- Klasické\* databáze nabízejí slabé, těžkopádné prostředky

\* hierarchické, síťové, relační, objektové

# 3. soustředění

# Návrh a provoz databází

# *Projekt IS*

- Rozeznání potřeby
  - specifikace požadavků I
- Rozhodnutí o cestě realizace
  - vlastními silami
  - zakázkou IT firmě
- Volba řešení
  - vývoj na míru
  - využití hotového typového produktu

# Vývoj IS

- Analýza požadavků II
  - funkcionalita
  - časové požadavky
  - informační požadavky
- Zjištění omezení
  - finanční, časové, lidské zdroje, prostorové...
  - business pravidla
- Modelování

# *Modelování IS*

- Klasické strukturované
  - procesní/funkční část
  - informační/datová část
  - business pravidla
- Moderní objektově orientované
  - use case
  - class diagram
  - sequence diagram
  - ...

# *Design IS*

- Design dat
- Design funkcí/aplikací
- Design nasazení
- Design bezpečnosti
- Design budoucí správy, zodpovědností
- ...



# *Nasazení a provoz IS*

- Správa, servis problémů
- Sběr a správa připomínek
- Doladování
- Doprogramování
- Úpravy v rámci navržené koncepce
  
- Požadavky přesahující rámec současné koncepce...?

# *Nový projekt*

- Nová analýza...
- Mezi omezeními je požadavek na hospodárné využití stávajících investic a stávajících systémů

# *Datová dimenze IS*

- 1 Úvodní analýza, požadavky
- 2 Design
- 3 Nové požadavky
- 4 Rozšíření?
  - Bez zásadnějších problémů
- 5 Úpravy?
  - Radikální změny představovaly a představují zásadní obtíže!

# *Datový model*

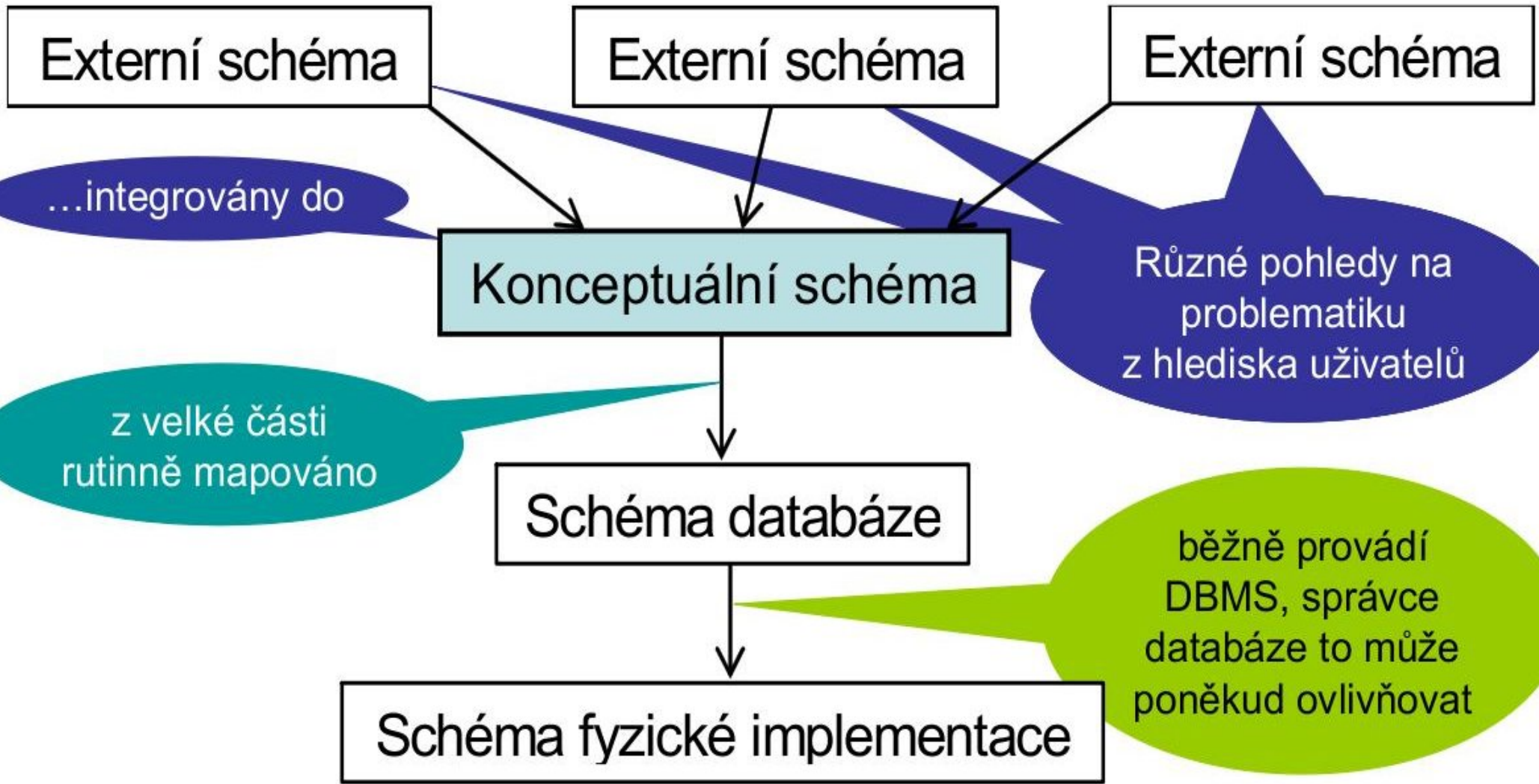
- Jak dosáhnout stability struktur dat pro danou oblast zájmu?
  - Oblast zájmu se také může rozšířit...
- Osvědčilo se: věrnost realitě
  - protože nové požadavky jsou o stejném světě

# *Princip tří schémat (ANSI/SPARC 1975)*

- Externí schéma
  - pohled uživatelů
- Analytický, konceptuální model
  - model světa
- Interní schéma
  - datové struktury

# *Postup shora dolů*

- Externí schémata
  - pohledy uživatelů
- Analytický, konceptuální model
  - model světa
- Logický model designu
  - bez rozhodnutí o implementačních technologiích
- Implementační schéma
  - volby v rámci implementačních technologií



Externí schéma

Externí schéma

Externí schéma

...integrovány do

Konceptuální schéma

Různé pohledy na problematiku z hlediska uživatelů

z velké části rutinně mapováno

Schéma databáze

běžně provádí DBMS, správce databáze to může poněkud ovlivňovat

Schéma fyzické implementace

# *Konceptuální schéma*

- Srozumitelný a věcně správný model, na jehož základě bude databáze navržena
  - srozumitelný, přehledný
  - vše podstatné
  - věcně správný
  
  - konceptuální, nikoli technologický



# *Konceptuální schéma*

- Společný základ pro chápání objektů aplikace uživateli, analytikem, správcem databáze i programátory

# *Konceptuální schéma*

- Poskytuje dokumentaci:
  - k ověření správnosti analýzy
  - východisko pro seznámení se stávajícími databázovými strukturami
  - východisko při analýze nových požadavků

# ***Jak dělat konceptuální schéma?***

- Analyzovat uživatelské požadavky
- Analyzovat realitu

# *Jak analyzovat realitu?*

- Přístupy:
  - Entity a vztahy mezi nimi
  - Všechno jsou funkce
  - Objekty a jejich role
  - Objekty, třídy,... v objektovém přístupu

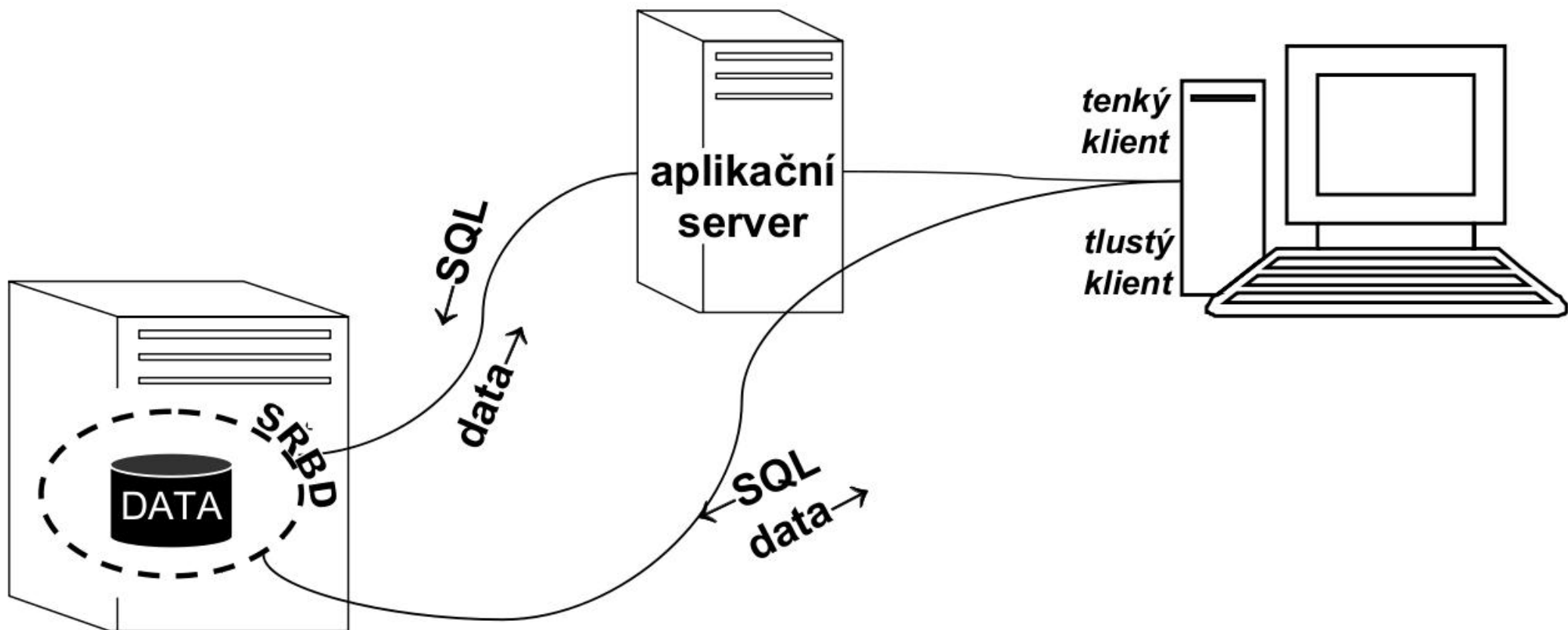
# ***Příklad***

Sledují se zápasy (kdo jsou domácí a kdo hosté, jaký je výsledek zápasu). Každý zápas píská 1 hlavní rozhodčí, 2 čároví, 1 náhradní rozhodčí. Eviduje se, za který tým hráči hrají (za jediný) tým v dané lize. Sleduje se skóre hráčů v zápasech. V dané lize dva týmy spolu hrají právě jednou jako domácí a právě jednou jako hosté.

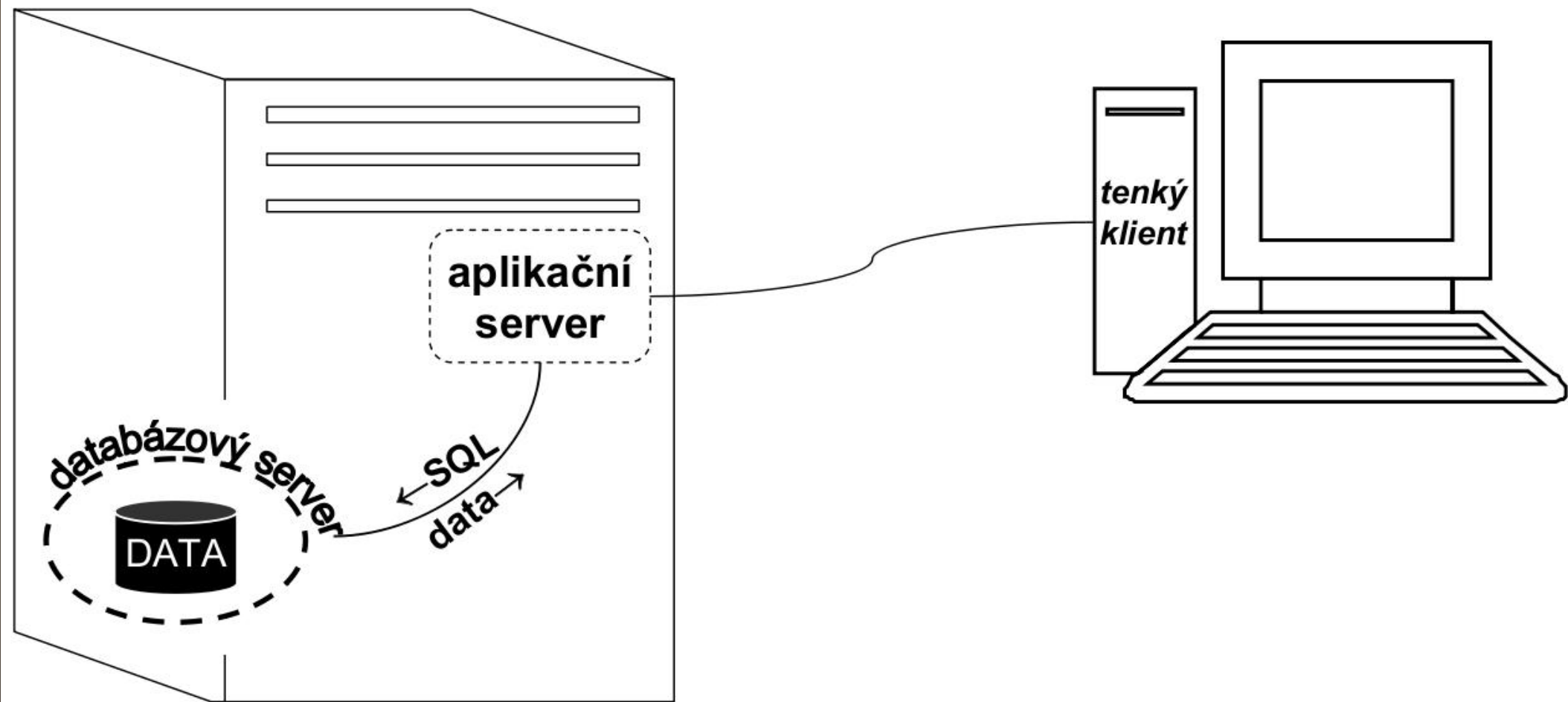
# *Datové modelování - podklady*

- <http://krokodata.vse.cz/DM>
- <http://nb.vse.cz/~palovska/4it218/KoncMod.zip>
- <http://nb.vse.cz/~palovska/Sa321/MetodaInformacniAnalyzy.zip>

# Nasazení a provoz databáze

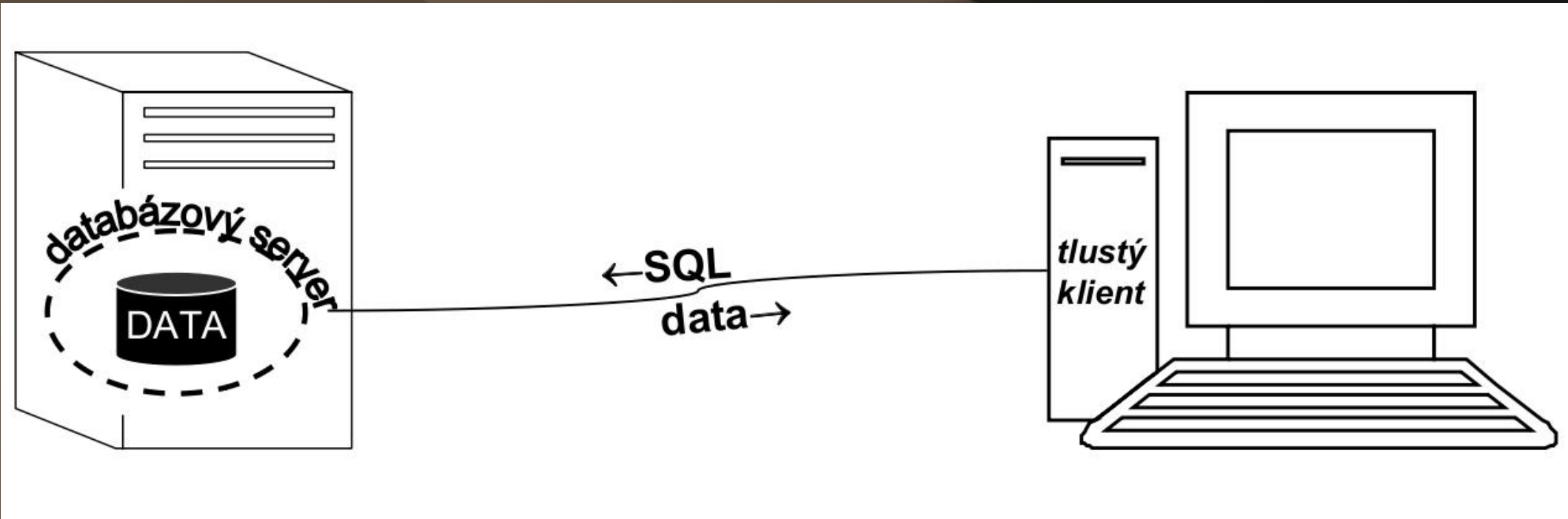


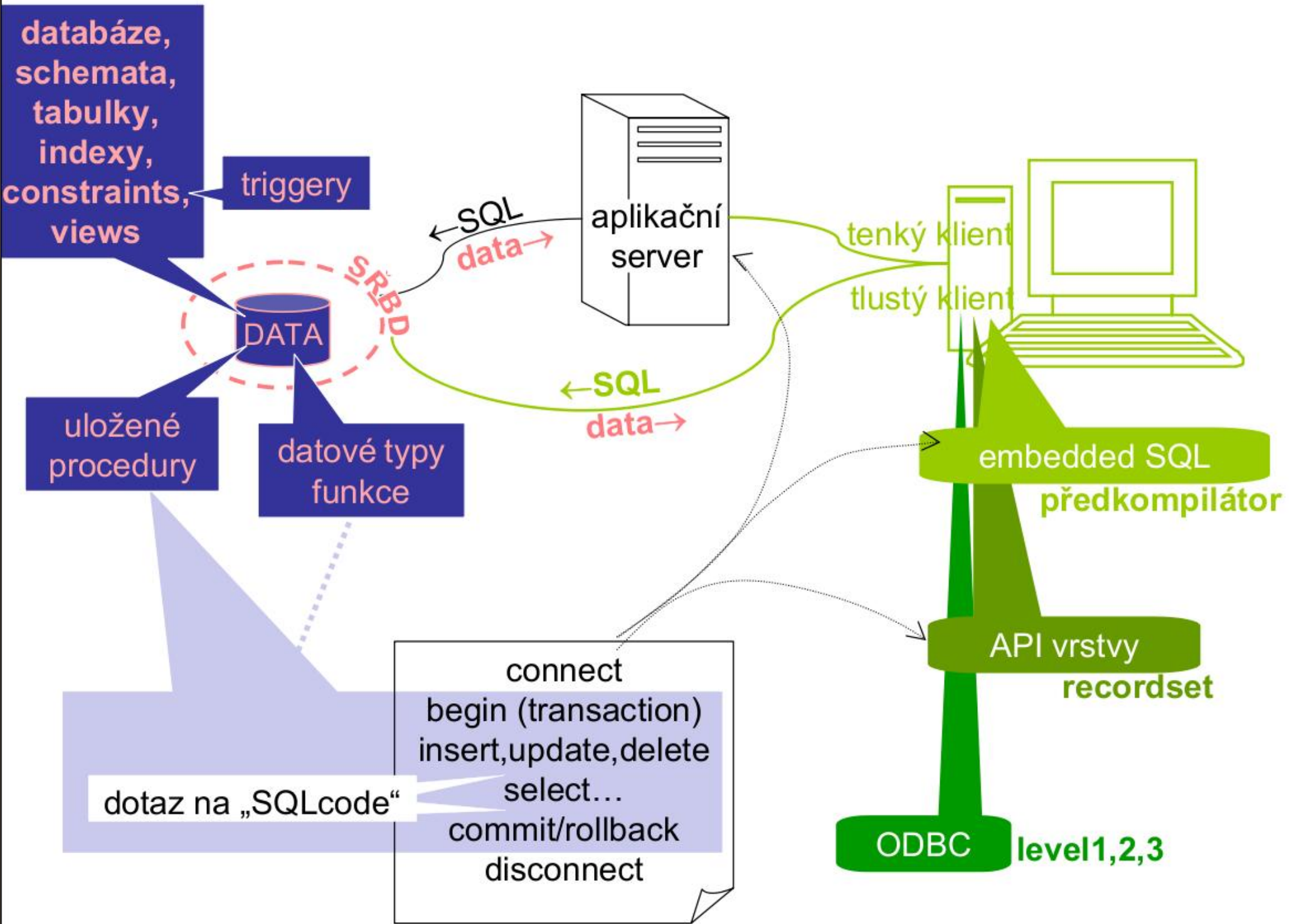
# Nasazení databáze, jiná varianta



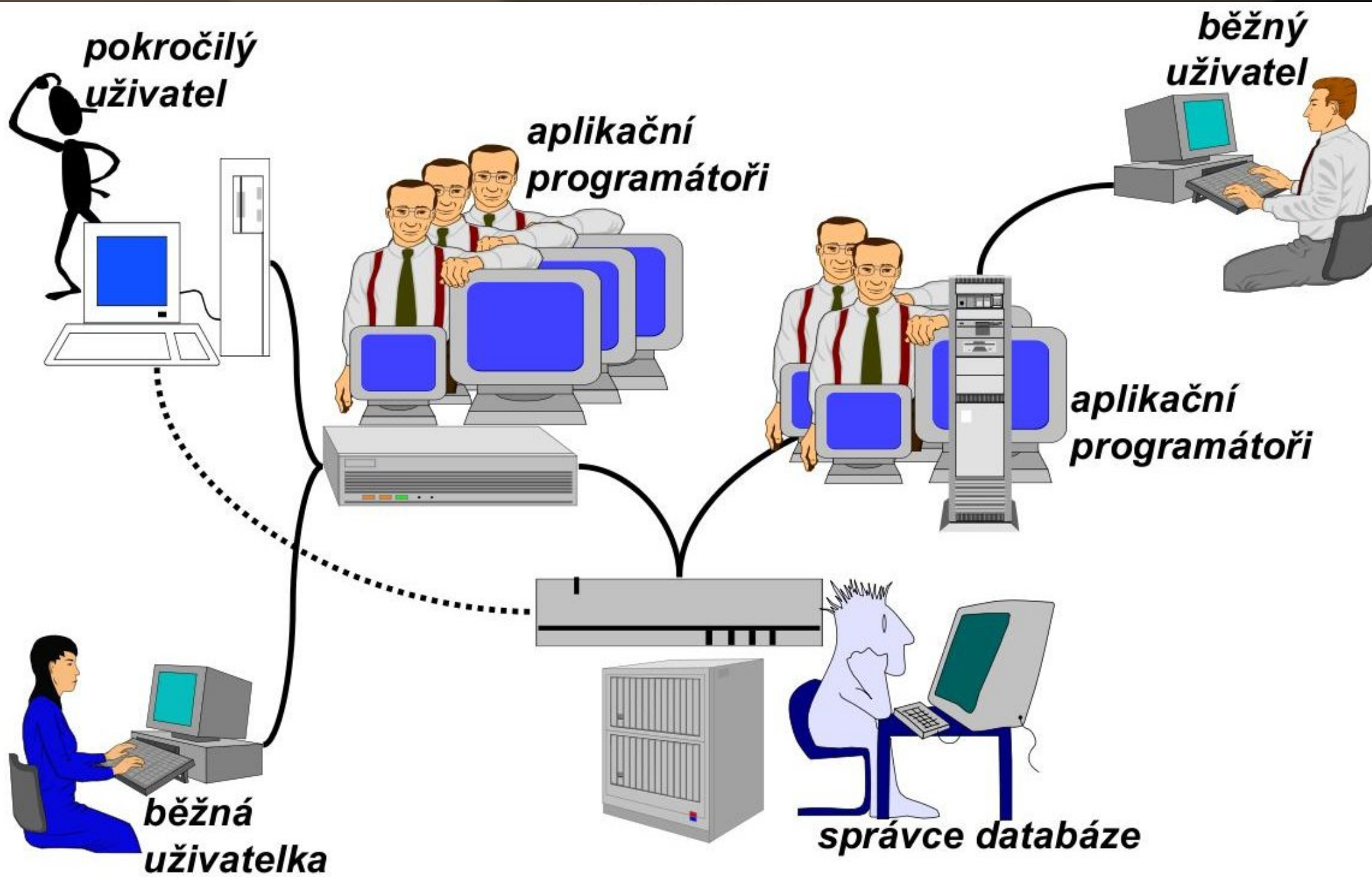


# Klientská „databáze“





# Provoz databáze



# *Správce databáze*

- Zná, které aplikace potřebují jaká data
  - spravuje bezpečnost
- Udržuje design tak, aby bylo možno rozvíjet
- Reaguje na provozní výkonové požadavky
  - radí programátorům při optimalizaci
  - optimalizuje prováděcí plány dotazů
  - spolupracuje na uložených procedurách
  - upravuje fyzické struktury
  - nasazuje další hw prostředky
  - nasazuje další db enginey

# *Nový projekt IS a databáze*

- Záhodná je informační kontinuita
  - nadále udržuje správný návrh dat
  - export a import dat
  - zálohy...
- Řešení přechodové doby
  - paralelní provoz starých a nových systémů
    - zajišťování konzistence informací